



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1459  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/697,184	10/30/2003	Asaf Adi	IL920030025US1	8188
7590 Stephen C. Kaufman Intellectual Property Law Dpt. IBM Corporation P. O. Box 218 Yorktown Heights, NY 10598				
EXAMINER				
CHUMPTIAZ, BOB R				
ART UNIT		PAPER NUMBER		
4115				
MAIL DATE		DELIVERY MODE		
07/25/2008		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/697,184

**Applicant(s)**

ADI ET AL.

**Examiner**

BOB CHUMPITAZ

**Art Unit**

4115

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 30 October 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-26 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-26 is/are rejected.
- 7) ☒ Claim(s) 20,23 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 30 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-8508)
- \_\_\_\_\_ Paper No(s)/Mail Date \_\_\_\_\_

- 4) ☐ Interview Summary (PTO-413)
- \_\_\_\_\_ Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

The following is a Non-Final First Office Action in response to the Patent Application filed October 30, 2003. Claims 1-26, as originally filed are presented for examination on the merits.

#### ***Claim Objections***

Claims 20, 23 are objected to because of the following informalities.

Claim 20 is objected to under 37 CFR 1.75 as being a substantial duplicate of claim 19.

Claim 23 is objected to for the use of undefined acronyms: "APIs". Claims should not contain acronyms that have not been previously defined in a claim. Appropriate correction is required.

#### ***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim 13 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 13 recites a system comprising "a model generator", and "an active dependency integration unit"; however as presented in the claims it is not directed to any form of structure. This subject matter is not limited to that which falls within a statutory category of invention because it is limited to a process, machine, manufacture, or a composition of matter. The claim recites function descriptive material and a function descriptive material is non-statutory subject matter.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 22 and 23 are rejected under 35 U.S.C 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 22 and 23, it recites the limitation: “wherein the model”. Claims 22 and 23 depends on claim 13. It is unclear whether applicant intends the term to refer to “dependency model” or “model generator”. For purposes of examination, Examiner will consider “model” to mean dependency model.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

**Claims 11 and 12 are rejected under 35 U.S.C. 102(b, e) as being anticipated by James Bailey, Alexander Poulouvassilis and Peter T. Wood (hereinafter referred as Bailey or Srivastava (US 2004/0128645 A1) or Levinson (US 6,381,580 B1).**

**As per claim 11**, Bailey or Srivastava or Levinson disclose a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for automatically updating business components information, and propagating changes in business components to other business components according to a dependency model, said method comprising:

applying rules that describe how an event affects a business component and that describe when a change in a business component triggers an event to create a business dependency model modeling business components and dependencies between them including compound dependencies, said dependency model defining event types, business component types, and dependency types associated with a business domain, said dependency model further defining how information is propagated from one business component to another (Bailey: Pg. 486, col. 2 ECA rules have been used in many setting including active databases....analysis techniques developed for conventional active databases can be applied in an XML setting; see also, Pg. 487, col. 1 an XML database consists of a set of XML documents; Srivastava: ¶ [0105-0114 computer readable program code...storage device...software may be stored in a computer readable medium]; Levinson: col. 6, line 6 –col. 7, line 13 read only memory, hard disk storage...software application program being executed by the CPU and the application program and data may be stored in the memory); and

responsive to one or more events and/or and constraints violations, automatically updating business components information, and propagating changes in business components to other business components according to the dependency model (Bailey: Pg. 486, col. 2 ECA rules have been used in many setting including active databases....analysis techniques developed

for conventional active databases can be applied in an XML setting; see also, Pg. 487, col. 1 an XML database consists of a set of XML documents; Srivastava: ¶ [0105-0114 computer readable program code...storage device...software may be stored in a computer readable medium]; Levinson: col. 6, line 6 –col. 7, line 13 read only memory, hard disk storage...software application program being executed by the CPU and the application program and data may be stored in the memory).

**As per claim 12**, Bailey or Srivastava or Levinson disclose a computer program product comprising a computer useable medium having computer readable program code embodied therein for automatically updating business components information, and propagating changes in business components to other business components according to a dependency model, said computer program product comprising:

computer readable program code for causing the computer to apply rules that describe how an event affects a business component and that describe when a change in a business component triggers an event to create a business dependency model modeling business components and dependencies between them including compound dependencies, said dependency model defining event types, business component types, and dependency types associated with a business domain, said dependency model further defining how information is propagated from one business component to another (Bailey: Pg. 486, col. 2 ECA rules have been used in many setting including active databases....analysis techniques developed for conventional active databases can be applied in an XML setting; see also, Pg. 487, col. 1 an XML database consists of a set of XML documents; Srivastava: ¶ [0105-0114 computer readable program code...storage device...software may be stored in a computer readable medium];

Art Unit: 4115

Levinson: col. 6, line 6 –col. 7, line 13 read only memory, hard disk storage...software application program being executed by the CPU and the application program and data may be stored in the memory); and

computer readable program code for causing the computer to automatically update business components information, and to propagate changes in business components to other business components according to the dependency model in response to one or more events and/or and constraints violations (Bailey: Pg. 486, col. 2 ECA rules have been used in many setting including active databases....analysis techniques developed for conventional active databases can be applied in an XML setting; see also, Pg. 487, col. 1 an XML database consists of a set of XML documents; Srivastava: ¶ [0105-0114 computer readable program code...storage device...software may be stored in a computer readable medium]; Levinson: col. 6, line 6 –col. 7, line 13 read only memory, hard disk storage...software application program being executed by the CPU and the application program and data may be stored in the memory).

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**Claims 1-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bailey or Huaxin Zhang and Eleni Stroulia (hereafter referred as Zhang) in view of Srivastava and in further view of Levinson.**

**As per claim 1**, Bailey or Zhang disclose a computer-implemented method for automatically updating business components information, and propagating changes in business components to other business components according to a dependency model, said method comprising:

Bailey or Zhang disclose applying rules that describe how an event affects a business component and that describe when a change in a business component triggers an event to create a business dependency model modeling business components and dependencies between them including compound dependencies (Bailey: Pg. 486, col. 2 – Pg. 487, col.1 Event-condition-action (ECA) rules automatically perform actions in response to events...ECA used in active databases, workflow management, network management, personalization, and specifying and implementing business processes....multiple ECA rules are defined with a system...the execution of one rule may cause an event which triggers another rule or set of rules; see also, Pg. 494, col. 1-2 techniques for analyzing the triggering and activation dependencies.....ECA rules have the a simple syntax and are automatically invoked in response to events....the specification of such events is part of the document object model...simple execution model of ECA rules; Zhang: pg. 285, ¶[2.0-2.2.2 X-logic generator....reactive integration....the event-condition-action (ECA) rule paradigm was developed in the context of active-databases research and was originally intended to support daemons monitoring transactions of interest and triggering further transactions in response....the Babel rule is an event-condition-action triple that specifies what task-initiation event should be invoked....Babel rules are specified in XML...rules are actually represented as XSLT program]),),



Bailey discloses said dependency model defining event types, business component types, and dependency types associated with a business domain (pg. 486, col.2 ECA rules used for many settings, including active databases, workflow management, network management specifying and implementing business processes; see also, pg 487, col. 1 the use of XPath and XQuery languages to specify events, conditions and actions within ECA rules; see also, pg. 494, col. 1 technique for analyzing the triggering and activation dependencies)

Bailey does not disclose said dependency model further defining how information is propagated from one business component to another; and

Srivastava teaches a model that includes one or more characteristics including the functionality of the component, the dependency of the component on other components, the performance of the component, the effort required to develop the component, and the effort required to integrate the component (§ [0021-0026 method includes the steps of generating representations of existing components, storing the representations for retrieval, formulating one or more issues arising in the project...the model includes the functionality of the component, the dependency of the component on other components, the performance of the component, the effort required to develop the component, and the effort required to integrate the component]);

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the execution model of ECA rules of Bailey to include a model as taught by Srivastava in order to provide a method, a system and a computer program product that can automatically identify and analyze different options and assist a user in making decisions relating to management of a project.

Bailey further discloses responsive to one or more events and/or and constraints violations (pg. 486, col. 2 ECA rules can be used for performing automatic repairs when violations are detected).

Bailey and Srivastava do not disclose automatically updating business components information.

Levinson teaches a system and method that adjusts or update a plan automatically as changes in the plan occur (col. 4, lines 14-57 the system may detect surprise and unanticipated events and adjust the plan to take into account those unanticipated events....as changes in the plan occur the floating appointments may be automatically moved around the user's schedule...method for automatically planning a series of events into a plan is provided; see also, col. 8, lines 46-55 the goal manager may update the goal list when needed; see also, col. 11, lines 13-25 once a changed calendar message has been received the goal manager may update the goals), and

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the execution model of ECA rules of Bailey and the model of Srivastava to include goal manager as taught by Levinson in order to provide the ability to reschedule events or appointments in response to surprises, distractions or unanticipated problems automatically, which will assist in detecting and resolving conflicts in the new schedule or business process.

Levinson further teaches propagating changes in business components to other business components according to the dependency model (col. 9, line 26 – col. 10, line 8 the planning and cueing system may generate a plan for the user based on updated information from the user...the

system may change the user's plan based on the user's responses or based on the unexpected event).

**As per claim 2**, Bailey further discloses further including receiving as input said event types, business component types, and dependency types associated with a business domain (pg. 486, col.2 ECA rules used for many settings, including active databases, workflow management, network management specifying and implementing business processes; see also, pg 487, col. 1 the use of XPath and XQuery languages to specify events, conditions and actions within ECA rules; see also, pg. 489 , col. 2 an implementation of XQuery to infer at query compile time the output type of a query on documents conforming to a given input type; see also, pg. 494, col. 1 technique for analyzing the triggering and activation dependencies).

**As per claim 3**, Bailey further disclose further including receiving as input rules that describe how an event affects a business component (Pg. 486, col. 2 – Pg. 487, col.1 Event-condition-action (ECA) rules automatically perform actions in response to events...ECA used in active databases, workflow management, network management, personalization, and specifying and implementing business processes....multiple ECA rules are defined with a system...the execution of one rule may cause an event which triggers another rule or set of rules; see also, pg. 489 , col. 2 an implementation of XQuery to infer at query compile time the output type of a query on documents conforming to a given input type; see also, Pg. 494, col. 1-2 techniques for analyzing the triggering and activation dependencies.....ECA rules have the a simple syntax and are automatically invoked in response to events...the specification of such events is part of the document object model...simple execution model of ECA rules).

**As per claim 4**, Bailey further discloses further including receiving as input rules that describe when a change in a business component triggers an event (pg. 486, col.2 ECA rules used for many settings, including active databases, workflow management, network management specifying and implementing business processes; see also, pg 487, col. 1 the use of XPath and XQuery languages to specify events, conditions and actions within ECA rules; see also, Pg. 489, col. 2 an implementation of XQuery to infer at query compile time the output type of a query on documents conforming to a given input type; see also, pg. 490, col. 1 the input to ECA rule execution is an XML database and a schedule; see also, pg. 494, col. 1 technique for analyzing the triggering and activation dependencies).

**As per claim 5**, Bailey further discloses further including defining said event types, business component types, and dependency types associated with a business domain (pg. 486, col.2 ECA rules used for many settings, including active databases, workflow management, network management specifying and implementing business processes; see also, pg 487, col. 1 the use of XPath and XQuery languages to specify events, conditions and actions within ECA rules; see also, Pg. 489, col. 2 an implementation of XQuery to infer at query compile time the output type of a query on documents conforming to a given input type; see also, pg. 490, col. 1 the input to ECA rule execution is an XML database and a schedule; see also, pg. 494, col. 1 technique for analyzing the triggering and activation dependencies).

**As per claim 6**, Bailey further discloses further including defining said that describe how an event affects a business component (Pg. 486, col. 2 – Pg. 487, col.1 Event-condition-action (ECA) rules automatically perform actions in response to events...ECA used in active databases, workflow management, network management, personalization, and specifying and implementing

business processes....multiple ECA rules are defined with a system...the execution of one rule may cause an event which triggers another rule or set of rules; see also, Pg. 494, col. 1-2 techniques for analyzing the triggering and activation dependencies.....ECA rules have the a simple syntax and are automatically invoked in response to events....the specification of such events is part of the document object model...simple execution model of ECA rules).

**As per claim 7**, Bailey further discloses further including defining said rules that describe when a change in a business component triggers an event (Pg. 486, col. 2 – Pg. 487, col.1 Event-condition-action (ECA) rules automatically perform actions in response to events...ECA used in active databases, workflow management, network management, personalization, and specifying and implementing business processes....multiple ECA rules are defined with a system...the execution of one rule may cause an event which triggers another rule or set of rules; see also, Pg. 494, col. 1-2 techniques for analyzing the triggering and activation dependencies.....ECA rules have the a simple syntax and are automatically invoked in response to events....the specification of such events is part of the document object model...simple execution model of ECA rules).

**As per claim 8**, Bailey further discloses wherein the business dependency model includes predefined dependency type semantics (Pg. 494, col. 2 method for computing rule triggering and activation relationships focus on determining the effects of update queries....query optimization strategies are possible...given a set of pre-defined queries, allow one to retain in memory only documents which are relevant to computing these queries...as updates are made....analyzing the effects of the updates on the collection of pre-defined queries).

**As per claim 9**, Bailey further discloses claim 8 as rejected above but does not disclose wherein said dependency type semantics include a “mandatory” logical operator that logically couples one or more source components of the dependency to one or more targets of the dependency and sets the targets to a worst state of the sources.

Srivastava further teaches a predicate which is a construct from logic that refers to the relationship between objects (§ [0051-0062, 0073, 0091, 0097-0099 information about software component, including preconditions (dependencies or inputs) and effects (functionalities or outputs), is represented by predicates....where the latter are represented by terms (constants or variables).....the functionality (i.e. affects) of a software component is represented by predicates.....predicate is a logical construct]).

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the query optimization strategies of Bailey to include predicates as taught by Srivastava in order to formulate one or more issues arising in the project as a planning problem using automated reasoning techniques to identify options relating to the project.

**As per claim 10**, Srivastava further teaches wherein said dependency type semantics include an “N out of M” logical operator that logically couples M source components of the dependency to one or more targets of the dependency and sets the targets to ok if at least N of the sources are ok and otherwise sets the targets to “fail”(§ [0052-0062, 0091 the state T is a collection of facts with the semantics that information corresponding to the predicates in the state holds....a planning problem is formulated and solved using algorithms....state-space planners are a type of planner algorithm that search in the space of possible plans (sequences of actions)]).

**Please Note:**

Applicant(s) are reminded that optional or conditional elements do not narrow the claims because they can always be omitted. See *e.g.* MPEP §2106 II C: “Language that suggest or makes optional but does not require steps to be performed or does not limit a claim to a particular structure does not limit the scope of a claim or claim limitation.

The method claims recite optional language and are highlighted in bold and italic. “As a matter of linguistic precision, optional elements do not narrow the claim because they can always be omitted.” *In re Johnston*, 435 F.3d 1381, 77 USPQ2d 1788, 1790 (Fed. Cir. 2006)(where the Federal Circuit affirmed the Board’s claim construction of “further including that said wall may be smooth, corrugated, or profiled with increased dimensional proportions as pipe size is increased” since “this additional content did not narrow the scope of the claim because these limitations are stated in the permissive form ‘may.’”).

**Claims 13-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bailey or Zhang in view of Srivastava and in further view of Levinson.**

**As per claim 13,** A system for automatically updating business components information, and propagating changes in business components to other business components according to a dependency model, said system comprising:

Bailey or Zhang disclose a model generator that applies rules that describe how an event affects a business component and that describe when a change in a business component triggers an event to create a business dependency model modeling business components and

Art Unit: 4115

dependencies between them including compound dependencies (Bailey: Pg. 486, col. 2 – Pg. 487, col.1 Event-condition-action (ECA) rules automatically perform actions in response to events...ECA used in active databases, workflow management, network management, personalization, and specifying and implementing business processes....multiple ECA rules are defined with a system...the execution of one rule may cause an event which triggers another rule or set of rules; see also, Pg. 494, col. 1-2 techniques for analyzing the triggering and activation dependencies.....ECA rules have the a simple syntax and are automatically invoked in response to events....the specification of such events is part of the document object model...simple execution model of ECA rules; Zhang: pg. 285, ¶[2.0-2.2.2 X-logic generator....reactive integration....the event-condition-action (ECA) rule paradigm was developed in the context of active-databases research and was originally intended to support daemons monitoring transactions of interest and triggering further transactions in response.....the Babel rule is an event-condition-action triple that specifies what task-initiation event should be invoked....Babel rules are specified in XML...rules are actually represented as XSLT program]),

Bailey discloses said dependency model defining event types, business component types, and dependency types associated with a business domain (pg. 486, col.2 ECA rules used for many settings, including active databases, workflow management, network management specifying and implementing business processes; see also, pg 487, col. 1 the use of XPath and XQuery languages to specify events, conditions and actions within ECA rules; see also, pg. 494, col. 1 technique for analyzing the triggering and activation dependencies) ,

Bailey does not disclose said dependency model further defining how information is propagated from one business component to another; and



Srivastava teaches a model that includes one or more characteristics including the functionality of the component, the dependency of the component on other components, the performance of the component, the effort required to develop the component, and the effort required to integrate the component (§ [0021-0026 method includes the steps of generating representations of existing components, storing the representations for retrieval, formulating one or more issues arising in the project...the model includes the functionality of the component, the dependency of the component on other components, the performance of the component, the effort required to develop the component, and the effort required to integrate the component]);

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the execution model of ECA rules of Bailey to include a model as taught by Srivastava in order to provide a method, a system and a computer program product that can automatically identify and analyze different options and assist a user in making decisions relating to management of a project.

Bailey further discloses an active dependency integration unit responsive to one or more events and/or constraints violations for automatically updating business components information (pg. 486, col. 2 ECA rules can be used for performing automatic repairs when violations are detected), and

Bailey and Srivastava do not disclose propagating changes in business components to other business components according to the dependency model.

Levinson teaches propagating changes in business components to other business components according to the dependency model (col. 9, line 26 – col. 10, line 8 the planning and

Art Unit: 4115

cuing system may generate a plan for the user based on updated information from the user...the system may change the user's plan based on the user's responses or based on the unexpected event).

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the execution model of ECA rules of Bailey and the model of Srivastava to include the planning and cuing system as taught by Levinson in order to provide the ability to reschedule events or appointments in response to surprises, distractions or unanticipated problems automatically, which will assist in detecting and resolving conflicts in the new schedule or business process.

**As per claim 14**, Bailey further discloses wherein the model generator includes an input for receiving said event types, business component types, and dependency types associated with a business domain (pg. 486, col.2 ECA rules used for many settings, including active databases, workflow management, network management specifying and implementing business processes; see also, pg 487, col. 1 the use of XPath and XQuery languages to specify events, conditions and actions within ECA rules; see also, pg. 489 , col. 2 an implementation of XQuery to infer at query compile time the output type of a query on documents conforming to a given input type; see also, pg. 494, col. 1 technique for analyzing the triggering and activation dependencies).

**As per claim 15**, Bailey further discloses wherein the model generator includes an input for receiving said rules that describe how an event affects a business component (Pg. 486, col. 2 – Pg. 487, col.1 Event-condition-action (ECA) rules automatically perform actions in response to events...ECA used in active databases, workflow management, network management,

personalization, and specifying and implementing business processes....multiple ECA rules are defined with a system...the execution of one rule may cause an event which triggers another rule or set of rules; see also, pg. 489 , col. 2 an implementation of XQuery to infer at query compile time the output type of a query on documents conforming to a given input type; see also, Pg. 494, col. 1-2 techniques for analyzing the triggering and activation dependencies.....ECA rules have the a simple syntax and are automatically invoked in response to events....the specification of such events is part of the document object model...simple execution model of ECA rules).

**As per claim 16**, Bailey further discloses wherein the model generator includes an input for receiving said rules that describe when a change in a business component triggers an event (pg. 486, col.2 ECA rules used for many settings, including active databases, workflow management, network management specifying and implementing business processes; see also, pg 487, col. 1 the use of XPath and XQuery languages to specify events, conditions and actions within ECA rules; see also, Pg. 489, col. 2 an implementation of XQuery to infer at query compile time the output type of a query on documents conforming to a given input type; see also, pg. 490, col. 1 the input to ECA rule execution is an XML database and a schedule; see also, pg. 494, col. 1 technique for analyzing the triggering and activation dependencies).

**As per claim 17**, Bailey and Levinson do not disclose wherein the model generator is coupled to a user interface for allowing manual input of said event types, business component types, and dependency types associated with a business domain.

Srivastava teaches a control module within a computer that is connected to other devices such as input/output interfaces via a bus that in turn can consists data, address, and control buses

(¶ [0105-0114 the method for automatically identifying options in a project can be implemented using a computer program product in conjunction with a computer system....the computer includes a control module, input/output (I/O) interfaces, a video interface....the control module is implemented to using a central processing unit that executes or runs a computer readable program code...user input to operate the computer can be provided by one or more of the input devices via the I/O interfaces; see fig. 8]).

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the execution model of ECA rules of Bailey and planning and cueing system of Levinson to include user interface as taught by Srivastava in order to provide a method, a system and a computer program product that allows for manual user input to generate event types, business types, and dependency types associated with a business domain.

**As per claim 18**, Srivastava further teaches wherein the model generator is coupled to a user interface for allowing manual input of said rules that describe how an event affects a business component (¶ [0105-0114 the method for automatically identifying options in a project can be implemented using a computer program product in conjunction with a computer system....the computer includes a control module, input/output (I/O) interfaces, a video interface....the control module is implemented to using a central processing unit that executes or runs a computer readable program code...user input to operate the computer can be provided by one or more of the input devices via the I/O interfaces; see fig. 8]).

**As per claim 19**, Srivastava further teaches wherein the model generator is coupled to a user interface for allowing manual input of said rules that describe when a change in a business

component triggers an event (§ [0105-0114 the method for automatically identifying options in a project can be implemented using a computer program product in conjunction with a computer system....the computer includes a control module, input/output (I/O) interfaces, a video interface....the control module is implemented to using a central processing unit that executes or runs a computer readable program code...user input to operate the computer can be provided by one or more of the input devices via the I/O interfaces; see fig. 8)).

**As per claim 20**, Srivastava further teaches wherein the model generator is coupled to a user interface for allowing manual input of said rules that describe when a change in a business component triggers an event (§ [0105-0114 the method for automatically identifying options in a project can be implemented using a computer program product in conjunction with a computer system....the computer includes a control module, input/output (I/O) interfaces, a video interface....the control module is implemented to using a central processing unit that executes or runs a computer readable program code...user input to operate the computer can be provided by one or more of the input devices via the I/O interfaces; see fig. 8)).

**As per claim 21**, Bailey and Levinson do not disclose wherein the active dependency integration unit is responsively coupled to a situation awareness unit for receiving therefrom new situations resulting from events triggered by a change in a business component.

Srivastava teaches the dependency of the component on other component which is included in the model's characteristic (§ [0016, 0021-0025, 0046, 0050, 0066, 0068, 0096, 0097 the model can include one or more characteristics including the functionality of the component....the dependency of the component on other component....effort required to

integrate the component....awareness and enhanced software components....user can be made aware of software components B whose enhancements are of possible interest while maintaining program A]; see also, Fig. 1-8 and associated text).

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the execution model of ECA rules of Bailey and the planning and cueing system of Levinson to include integration and awareness components as taught by Srivastava in order to solve a business problem by including the identification of existing components for reuse, the identification of existing components to be upgraded and the specification of new components to be developed and providing automated reasoning techniques like model checking and constraint satisfaction.

**As per claim 22**, Bailey further discloses wherein the model is an XML format (Pg. 486, col. 2 ECA rules can be used in as an integrating technology for providing this kind of reactive functionality on XML repositories.....analysis techniques developed in conventional active databases can be applied in an XML setting).

**As per claim 23**, Srivastava further teaches wherein the model is loaded on startup under control of suitable APIs (§ [0110-0112 the computer system includes the computer readable medium having such software or program code recorded such that instructions of the software or the program code can be carried out]).

**As per claim 24**, Bailey further discloses wherein the business dependency model includes predefined dependency type semantic (Pg. 494, col. 2 method for computing rule triggering and activation relationships focus on determining the effects of update queries....query

optimization strategies are possible...given a set of pre-defined queries, allow one to retain in memory only documents which are relevant to computing these queries...as updates are made....analyzing the effects of the updates on the collection of pre-defined queries).

**As per claim 25**, Bailey further discloses claim 24 as rejected above but does not disclose wherein said dependency type semantics include a "mandatory" logical operator that logically couples one or more source components of the dependency to one or more targets of the dependency and sets the targets to ok if all the sources are ok and sets the targets to a worst state of the sources.

Srivastava further teaches a predicate which is a construct from logic that refers to the relationship between objects (§ [0051-0062, 0073, 0091, 0097-0099 information about software component, including preconditions (dependencies or inputs) and effects (functionalities or outputs), is represented by predicates....where the latter are represented by terms (constants or variables)....the functionality (i.e. affects) of a software component is represented by predicates....predicate is a logical construct]).

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the query optimization strategies of Bailey to include predicates as taught by Srivastava in order to formulate one or more issues arising in the project as a planning problem using automated reasoning techniques to identify options relating to the project.

**As per claim 26**, Srivastava further teaches wherein said dependency type semantics include an "N out of M" logical operator that logically couples M source components of the dependency to one or more targets of the dependency and sets the targets to ok if at least N of the

sources are ok and otherwise sets the targets to "fail" (§ [0052-0062, 0091 the state T is a collection of facts with the semantics that information corresponding to the predicates in the state holds....a planning problem is formulated and solved using algorithms....state-space planners are a type of planner algorithm that search in the space of possible plans (sequences of actions)]).

**Please note:**

A recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim.

The apparatus and systems claims recite use language, i.e., claim 12 recites computer readable program code *for causing the computer to apply rules*. See e.g. *In re Collier*, 158 USPQ 266, 267 (CCPA 1968)(where the court interpreted the claimed phrase “a connector member for engaging shield means” and held that the shield means was not a positive element of the claim since “[t]here is no positive inclusion of ‘shield means’ in what is apparently intended to be a claim to structure consisting of a combination of elements.”

Examiner has pointed out particular references contained in the prior arts of record in the body of this action for the convenience of the applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant, in preparing the response, to consider fully the entire references as



potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior arts or disclosed by the examiner.

### ***Conclusion***

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Moore et al. (5,446,885) Event driven management information system with rule-based applications structure stored in a relational database.
- Parad (5,369,570) Method and system for continuous integrated resource management.
- Chen (US 2004/0024767 A1) Method and System for managing event information in a computer network.
- Triron et al. (US 6,983,321 B2) System and method of enterprise systems and business impact management.
- Botzer et al. (US 2003/0204491 A1) Method and system for ensuring system awareness with data base connection on demand.
- Eric S.K. Yu and John Mylopoulos: AI models for business process reengineering. (IEEE © 1996).
- James Bailey and Alexander Poulouvassilis: An abstract interpretation framework for termination analysis of active rules. (Springer-Verlag Berlin Heidelberg © 2000).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to BOB CHUMPITAZ whose telephone number is (571)270-5494. The examiner can normally be reached on M-TR: 7:30 AM - 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, BRADLEY BAYAT can be reached on (571) 272-6704. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

B. C.  
Examiner, Art Unit 4115

/Bradley B Bayat/  
Supervisory Patent Examiner, Art Unit 4115